Image Processing - CS 474
Programming Assignment 3
Anthony Silva
November 10, 2025

“I declare that all material in this assignment is my own work except where there is clear acknowledgment or reference to the work of others. I understand that both my report and code may be subjected to plagiarism detection software, and fully accept all consequences if found responsible for plagiarism, as explained in the syllabus, and described in UNR’s Academic Standards Policy: UAM 6,502.”

- Anthony Silva

Theory

Introduction

The Discrete Fourier Transform (DFT) is a fundamental tool in image processing that transforms a function defined in the spatial domain into a frequency domain representation. This transformation represents the frequency components present in a given input allowing for new ways of transformations and analysis.

1D DFT

The forward 1-D DFT transforms a sequence of N samples f(x) into the frequency domain F(u):

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N}, \ u = 0, 1, 2, \ldots, N - 1$$

The inverse DFT reconstructs the function in the spatial domain from its frequency domain representation:

$$f(x) = \sum_{u=0}^{N-1} f(x)e^{2\pi ux/N}, \ x = 0, 1, 2, \ldots, N - 1$$

Values produced from the DFT are complex and can be represented in rectangular form with its real and imaginary components or in polar form with magnitude and phase components.

$$F(u) = R(u) + jI(u) = |F(u)|e^{j\phi(u)}$$

Where

$|F(u)| = \sqrt{R(u)^2 + I(u)^2}$

$\phi(u) = arctan(\frac{I(u)}{R(u)})$

2D DFT

The 2-D DFT extends the 1-D transform into two dimensions, making it applicable to images. The forward 2-D DFT of an image f(x,y) of size M x N is defined as:

$$F(u, v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)e^{-j2\pi(ux/M + vy/N)}, \ for\, u = 0, 1, 2, \ldots, M - 1 \ and\, v = 0, 1, 2, \ldots, N - 1$$

The inverse 2-D DFT reconstructs the spatial domain image from the frequency domain:

$$f(x,\ y)\ =\ \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,\ v)e^{j2\pi(ux/M\ +\ vy/N)},\ for\ x\ =\ 0, 1, 2\ ...,\ M\ -\ 1\ and\ y\ =\ 0, 1, 2,\ ...,\ N\ -\ 1$$

Properties

Several important properties of the Fourier Transform are explored in this report:

- Separability: The 2-D DFT can be computed by taking 1-D DFTs along rows, followed by 1-D DFTs along the columns of the result. This is due to the separability of the 2D exponential kernel into separate transforms. This allows for the algorithm to be calculated in a more efficient manner, and when calculated in a dynamic programming method, the complexity is reduced even more.
- Translation: Multiplying the spatial domain function by $(-\ 1)^{x+y}$ shifts the zero-frequency component to the center of the frequency domain through the following relationship: $f(x,\ y)(-\ 1)^{x+y}\ \ <=>\ F(u\ -\ N/2,\ v\ -\ M/2)$. This is useful for visualization as it places the central frequency at the center instead of at the beginning.
- Scaling: As the spatial extent of an object increases, its frequency domain representation becomes more concentrated. Similarly, smaller spatial extents create wider frequency representations.
- Magnitude and Phase: The magnitude spectrum represents the strength of each frequency component while the phase encodes the spatial relationships between different frequency components and preserves edge locations and structural details.
- Logarithmic Scaling: The Fourier Transform's magnitudes have a large range while values for display are typically only in the initial range. For that reason the logarithmic transformation of $log(1\ +\ |F(u, v)|)$ is used to compress the range and enhance the visibility of the smaller magnitude components (which are more common).

Results

1D Fourier Transform

To test my implementation of the Discrete Fourier Transform, I started by testing three fundamental 1D functions, computing their Fourier transforms, and verifying alignment with theoretical properties. The results are shown below in Table A. The graphs shown are scaled to the max and min values of the values.

The DFT of the sequence [2, 3, 4, 4] was computed and yielded:

Real Values: [13, -2, -1, -2]
Imaginary Values: [0, -1, 0, 1]
Inverse DFT: [2, 3, 4, 4]

These results match the theoretical expectation of [13, -2-i, -1, -2+i]. The successful reconstruction of the original sequence through the inverse DFT confirms the correctness of the implementation. The DC component (first value) of 13 equals the sum of the input sequence divided by N, representing the average value scaled by the sequence length.

A cosine wave with 8 complete cycles over 128 samples was generated using $f(x)\ = cos(2\pi ux/N) f(x)\ =\ cos(2\pi piux/N)$ where u = 8 and N = 128.

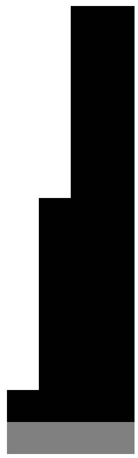
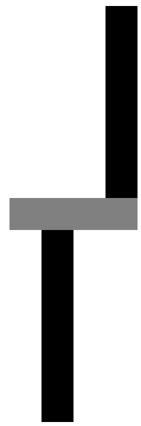
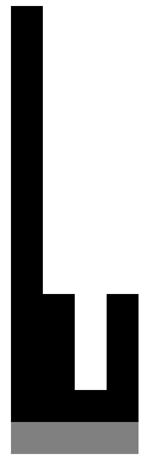
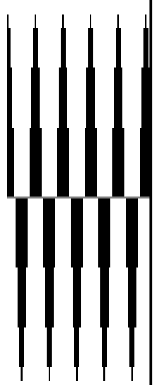
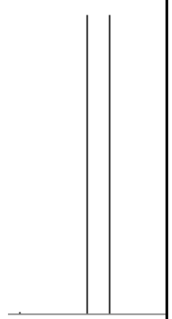
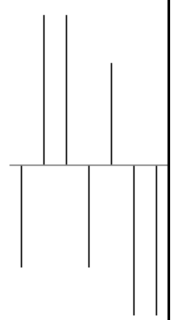
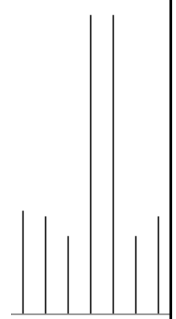
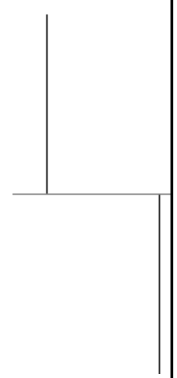
After applying the centering property $(-\ 1)^x$, the magnitude spectrum shows two distinct impulses at indices 56 and 72, which correspond to frequencies at -8 and +8 cycles relative to the center (index 64). This symmetric pair of impulses is the expected result for a pure cosine wave, as the cosine function consists of two exponential components at positive and negative frequencies.

The statistics confirm this behavior:

Maximum magnitude values of 8115.27 appear at the expected frequency locations and imaginary components are near zero. The real part dominates, consistent with the even symmetry of the cosine function

The rectangle function (width of 64, centered) produces a sinc function in the frequency domain, as predicted by fourier theory. The real part shows this sinc pattern with a strong maximum magnitude with alternating positive and negative spikes creating the sinc ripple and decreasing the amplitude as the frequency increases. The imaginary component is essentially zero throughout (max magnitude of 255 compared to the max real magnitude of 16320) which confirms the symmetry of the input signal. Scaling the values to fit within a set image resolution causes some visual awkwardness in the distributions.

Experiment	Initial	Real	Imaginary	Magnitude	Phase	Statistics

f = [2, 3, 4, 4]						N/A
Cosine Wave 8 periods in one unit of time 128 samples						Average Real: 127 Average Imag: -2.84217e-14 Max Real: 8115.27 Max Imag: -5.12007e-05 Values at index 8: Real: 8115.27 Imag: -3.46367e-05 Mag: 8115.27 Values at index 120: Real: 8115.27 Imag: 3.46367e-05 Mag: 8115.27

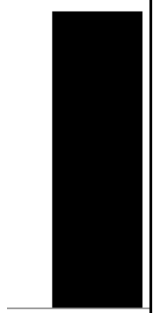
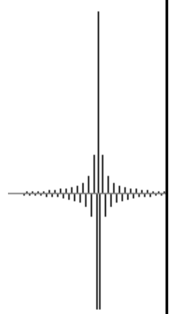
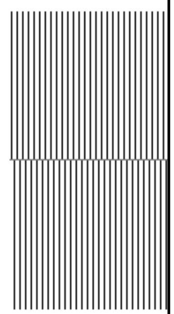
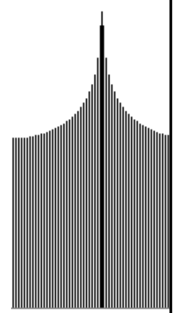
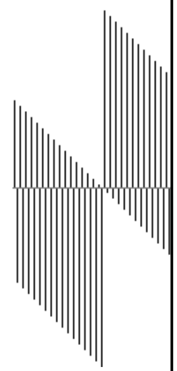
Rectangle Function width of 64 centered						Average Real: -7.27177e-06 Average Imag: 0 Max Real: 16320 Max Imag: -255

Table A. Showing functions and the real, imaginary, magnitude, and phase components of their fourier transform.

2D Fourier Transform

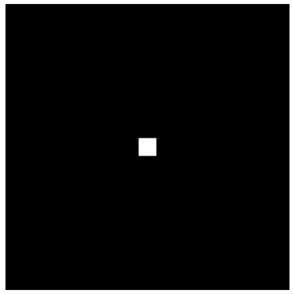
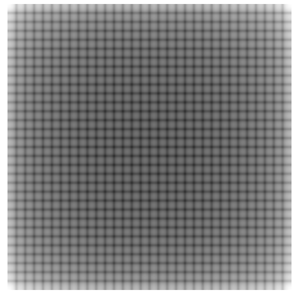
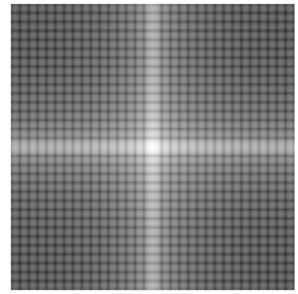
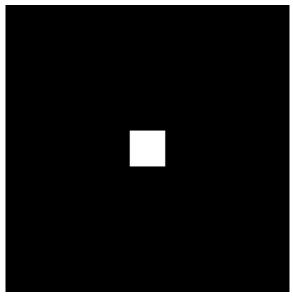
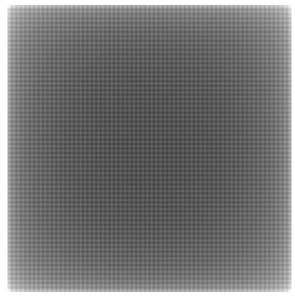
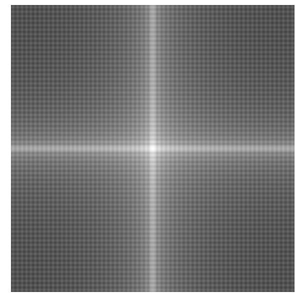
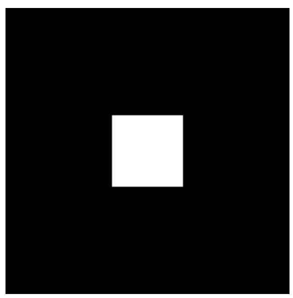
Three squares of different sizes (32x32, 64x64, 128x128) were centered in 512x512 images to examine the relationship between spatial size and frequency content.

The unshifted magnitudes for all three cases show the characteristic four-quadrant symmetry due to the periodic nature of the DFT. After applying the centering property using (-1)^(x+y), the shifted magnitudes reveal clear 2-D sinc function patterns centered in the frequency domain.

For the 32×32 square, the magnitude displays a bright central cross representing low-frequency content with symmetrically distributed side lobes extending relatively far from the center. The 64×64 square produces a more concentrated pattern as the central cross becomes narrower, with the center approximately halved compared to the 32×32 case. The 128×128 square shows the most concentrated frequency spectrum, with a very narrow and sharp central cross and sections compressed much closer to the center.

These differences demonstrate the inverse scaling property of the Fourier Transform: as the spatial extent increases by a factor of k, the frequency domain representation compresses by a factor of 1/k. This inverse relationship means that large, slowly varying spatial features correspond to low-frequency components, while small, rapidly changing features correspond to high frequencies. The 2-D sinc pattern remains consistent across all three cases, but its scale varies inversely with the square size.

Size	Rectangle	Magnitude (No Shift)	Magnitude (Shift)

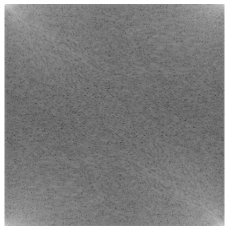
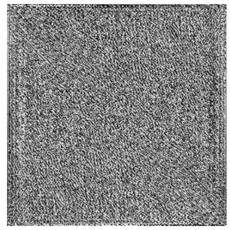
32			
64			
128			

Fourier Transform Component Image Reconstruction

This experiment examined the importance of magnitude and phase information for reconstructing an image.

When the phase was set to zero and only the magnitude information was used, the reconstructed image looked nothing like the original and rather resembled a small speck of light. When the magnitude was set uniformly across all frequencies while preserving the original phase, the reconstructed image retained the edge information of the original image, making the features of the original Lenna image clearly visible.

This makes sense as without phase information, the frequency components lack spatial alignment information. Phase then encodes the precise spatial alignment of different frequency components which allows for edges to become visible. With a uniform magnitude across the image, each frequency is equally important so there is no contrast.

Original Image	Magnitude	Phase	Magnitude Reconstructed	Phase Reconstructed
				

Conclusion

This assignment provided an opportunity to implement the DFT and explore its properties in both one and two dimensions.

The 1-D experiments confirmed theoretical predictions of simple functions and their expected frequency components after the fourier transform. The results validated the implementation through both visual inspection and numerical verification.

The 2-D square experiments showed the inverse relationship between spatial and frequency domains, when spatial features increase in size, their frequency representation becomes more concentrated.

The magnitude versus phase reconstruction experiment showed how phase provides edge information while magnitude carries pixel gradient and contrast values that apply to the image. Overall, phase was clearly more important for recreating an image.

Working directly with Fourier Transform implementations makes understanding its concepts easier and allows for an easier foundation for exploring modern image processing techniques that utilize the Fourier Transform. The implementation and visual results solidified theoretical concepts into practice, helping build intuition about the spatial and frequency domain relationships established through the Fourier Transform.